Tutorial: Optimizing Applications for Performance on POWER

Hardware and Software Overview

SCICOMP13

July 2007

Agenda

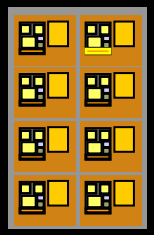
- Hardware
- Software
- Documentation

Hardware Overview

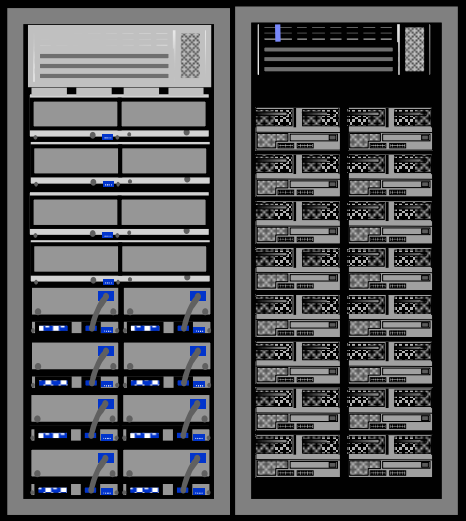
Core
"CPU"
"processor"

Chip
"socket"
"processor"

Node



Cluster

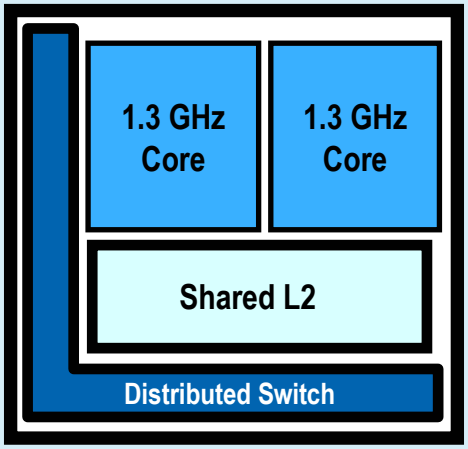


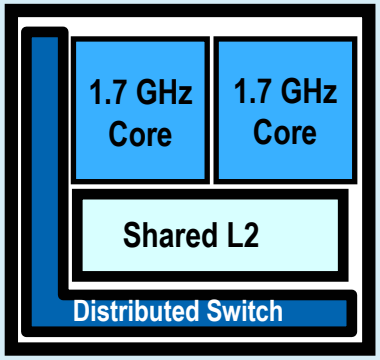
IBM Product Naming

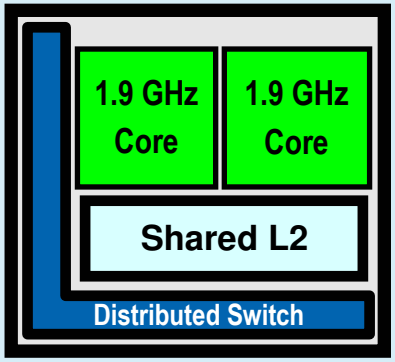
New Name	Old Names	Market	Processor
System i	iSeries, AS400	Commercial	RS64 POWER5
System p	RS6000 SP pSeries	Server, technical	POWER3 POWER4 POWER5 POWER5+ POWER6
System x	xSeries IA-32	Server, technical	Intel AMD PowerPC
System z	zSeries ES9000	Mainframe	zSeries

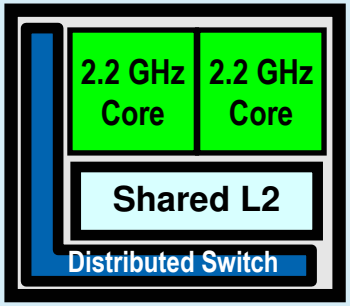
POWER Server Roadmap

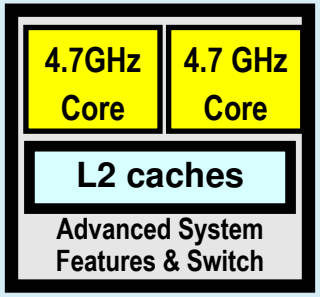
2001	2002-3	2004	2005-06	2007
POWER4	POWER4+	POWER5	POWER5+	POWER6
180 nm	130 nm	130 nm	90 nm	65 nm











- Chip Multi Processing
  - Distributed Switch
  - Shared L2
- Dynamic LPARs (16)

- Reduced size
- Lower power
- Larger L2
- More LPARs (32)

- Simultaneous multi-threading
- Sub-processor partitioning
- Dynamic firmware updates
- Enhanced scalability, parallelism
- High throughput performance
- Enhanced memory subsystem

- Ultra High Frequency
- Very Large L2
- Robust Error Recovery
- High ST and HPC Perf
- High throughput Perf
- More LPARs (1024)
- Enhanced memory subsystem

Autonomic Computing Enhancements

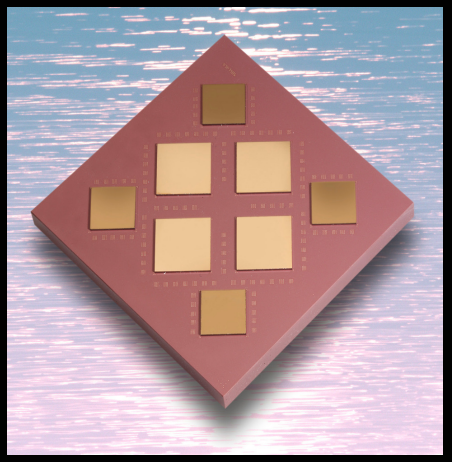
POWER5 Modules

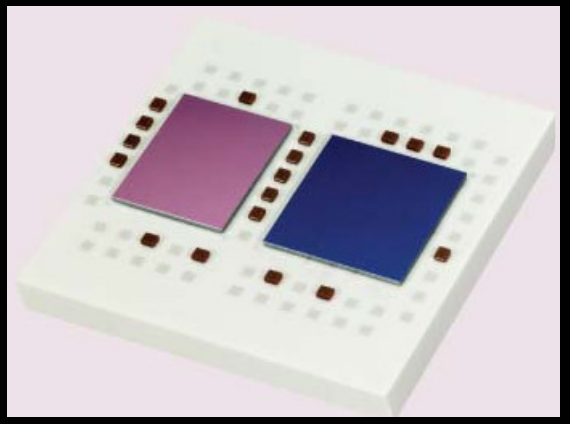
MCM

- 95mm × 95mm
- Four POWER5 chips
- Four cache chips
- 4,491 signal I/Os
- 89 layers of metal

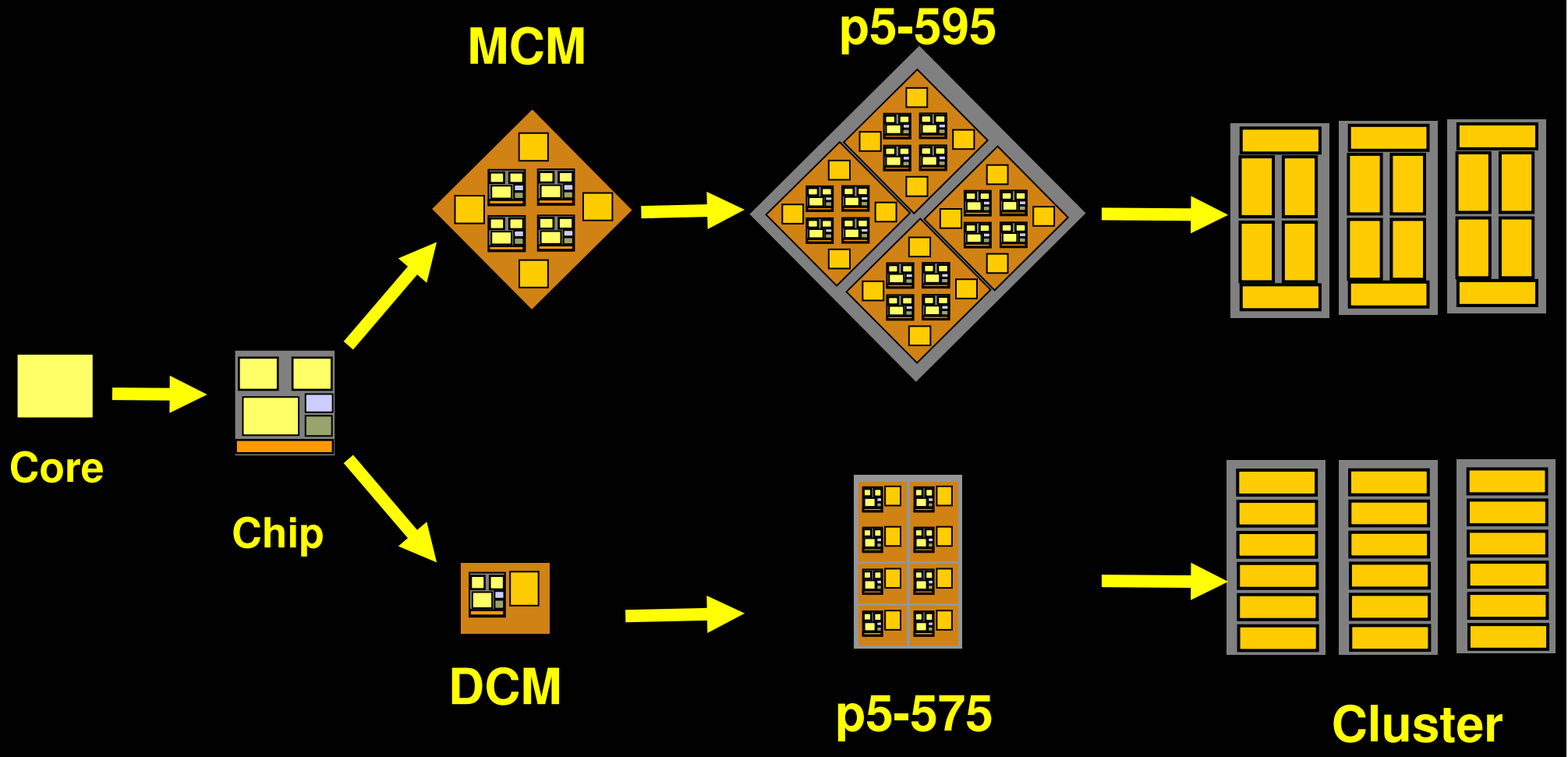
DCM

- One POWER5 chip
  - Single or Dual Core
- One L3 cache chip

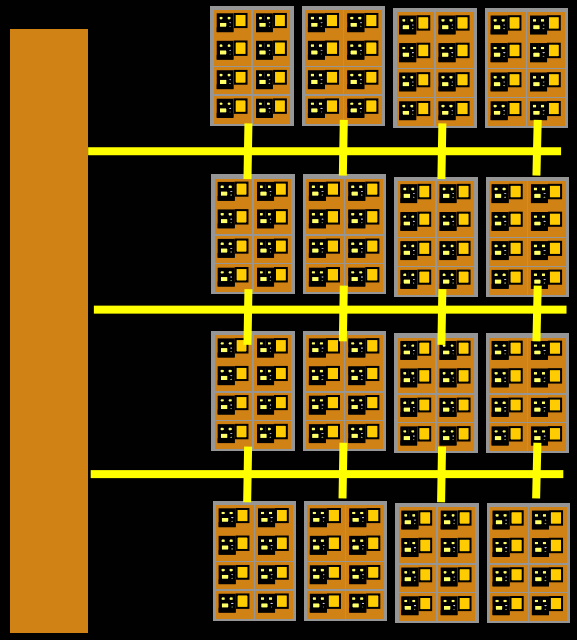




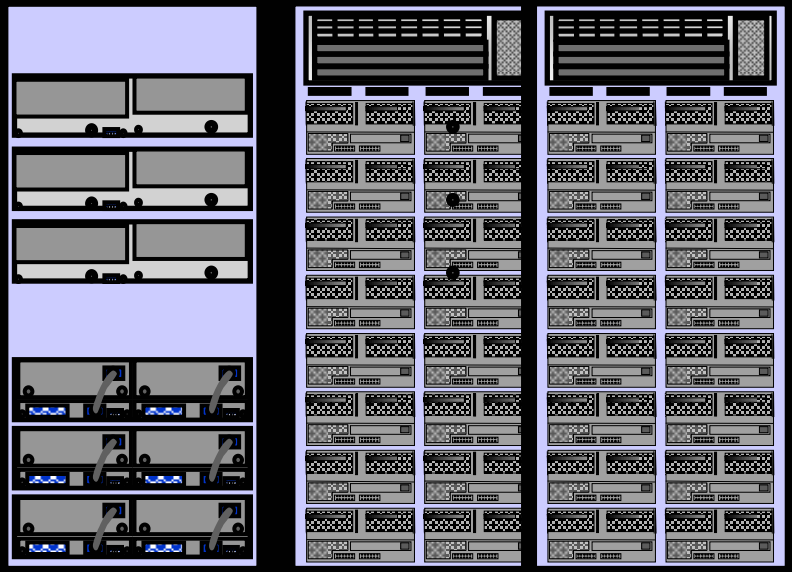
POWER5 Processor Systems



Cluster 1600



Logical View



Physical View

System p5 “Nodes” – partial list

Model	Processors	Clock Rate (GHz)	Max Memory (x 2^30 byte)
p5 595	16-64	1.65 - 2.3*	2000
p5 590	8-32	1.65,2.1*	1000
p5 575	8-16	1.9, 2.2*	256
p5 570	2-16	1.9, 2.2*	512
p5 560Q	4-16	1.5*	128
p5 520	1,2	1.65, 1.9*	32
p5 505	1,2	1.5, 1.65*	32

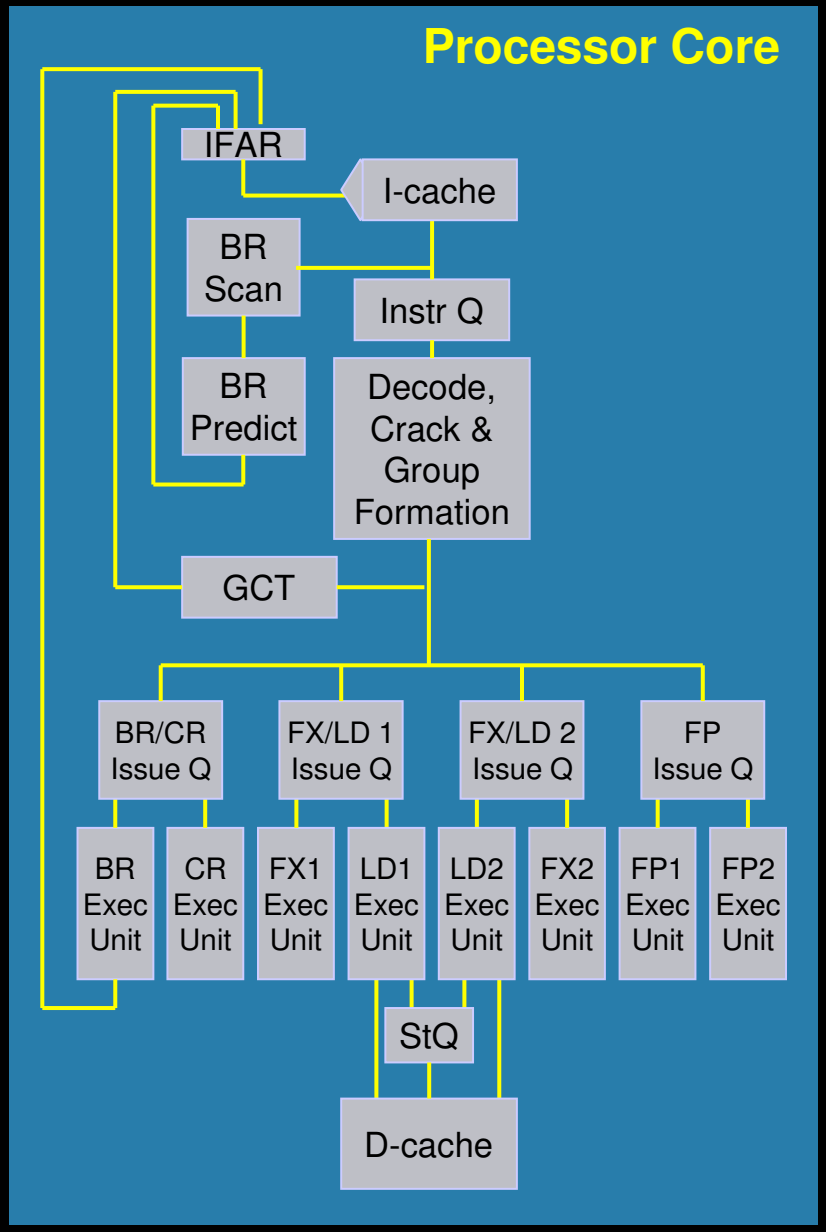
* - POWER5+

POWER5 Features

- Multi-processor
- Cache
  - Private L1 cache
  - Shared L2 cache
  - Shared L3 cache
- Interleaved memory
- Hardware Prefetch
- Multiple Page Size support
- Superscalar
- Speculative out-of-order instructions
- Up to 8 outstanding cache line misses
- Large number of instructions in flight
- Branch prediction

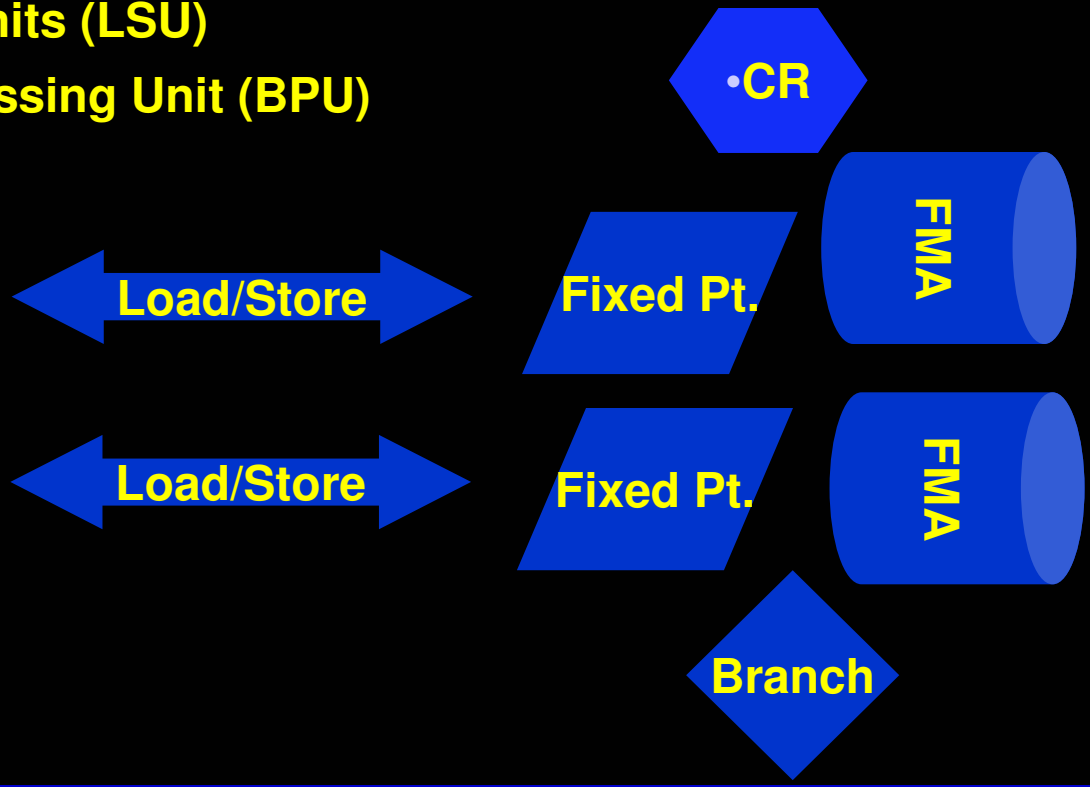
Instruction-level Parallelism

- Speculative superscalar organization
  - Out-of-Order execution
  - Large rename pools
  - 8 instruction issue, 5 instruction complete
  - Large instruction window for scheduling
- 8 Execution pipelines
  - 2 load / store units
  - 2 fixed point units
  - 2 DP multiply-add execution units
  - 1 branch resolution unit
  - 1 CR execution unit
- Aggressive branch prediction
  - Target address and outcome prediction
  - Static prediction / branch hints used
  - Fast, selective flush on branch mispredict



Multiple Functional Units

- Symmetric functional units
  - Two Floating Point Units (FPU)
  - Three Fixed Point Units (FXU)
    - Two Integer
    - One Control
  - Two Load/Store Units (LSU)
  - One Branch Processing Unit (BPU)



Register Renaming

- Architecture has 32 registers
  - Legacy
- Cases which require additional registers:
  - Tight loops
    - Computationally intensive
  - "Broad" loops
    - Many variables involved
  - Deep pipe lines
- Renaming registers are increasingly important with Simultaneous MultiThreading

Register Renaming

Read after write

$R_{13} = R_{14} + R_{15}$

...

$R_{16} = R_{13} + R_{12}$

Read after write

A(i) = B(i) + C(i)

....

D(i) = A(i) + E(i)

Write after write

$R_{13} = R_{14} + R_{15}$

...

$R_{13} = R_{16} + R_{17}$

$R_{19} = R_{13} + R_{18}$

Write after read

$R_{14} = R_{13} + R_{15}$

...

$R_{42} = R_{16} + R_{17}$

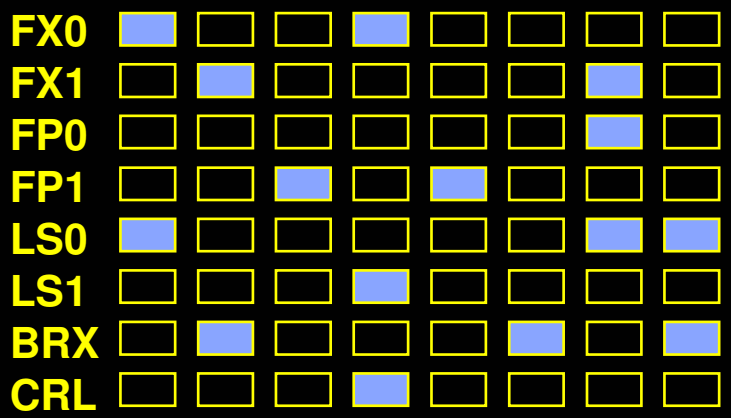
$R_{19} = R_{42} + R_{18}$

Effect of Registers

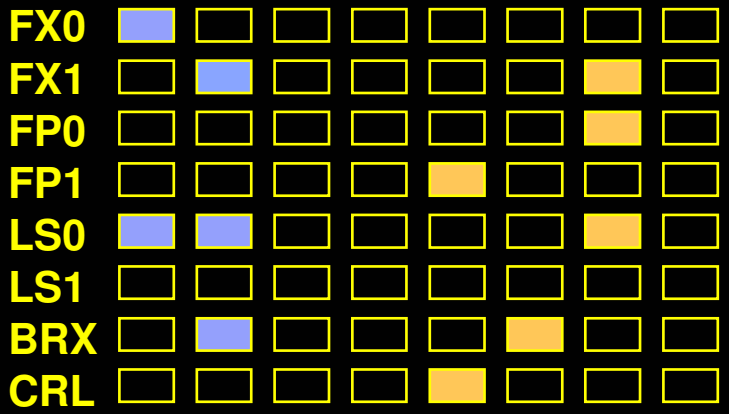
	POWER4	POWER5
GP Registers	80	120
FP Registers	72	120
DGEMM speed	60% of burst	90% of burst

Multi-threading Evolution

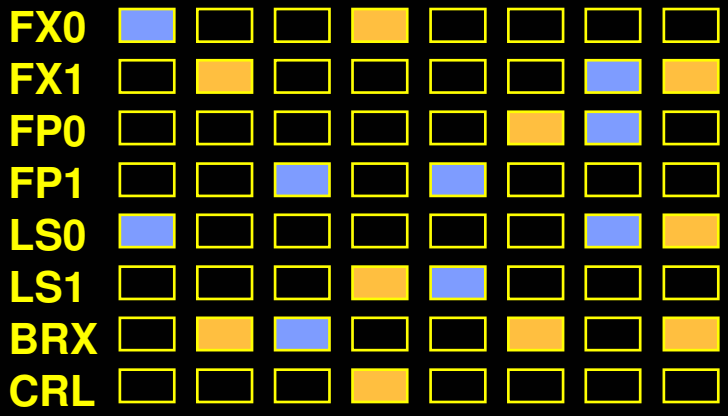
Single Thread



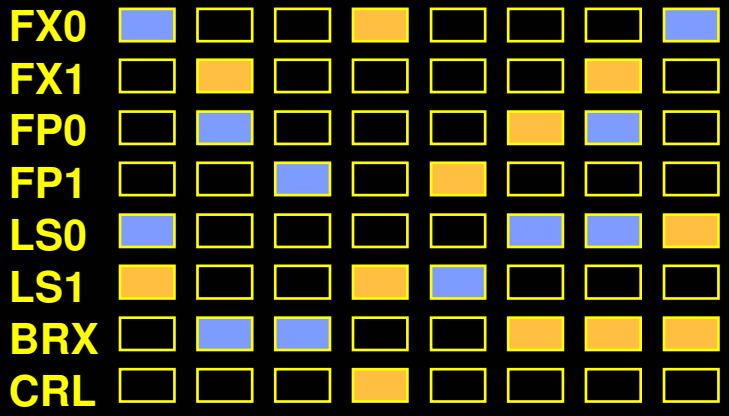
Coarse Grain Threading



Fine Grain Threading



Simultaneous Multi-Threading



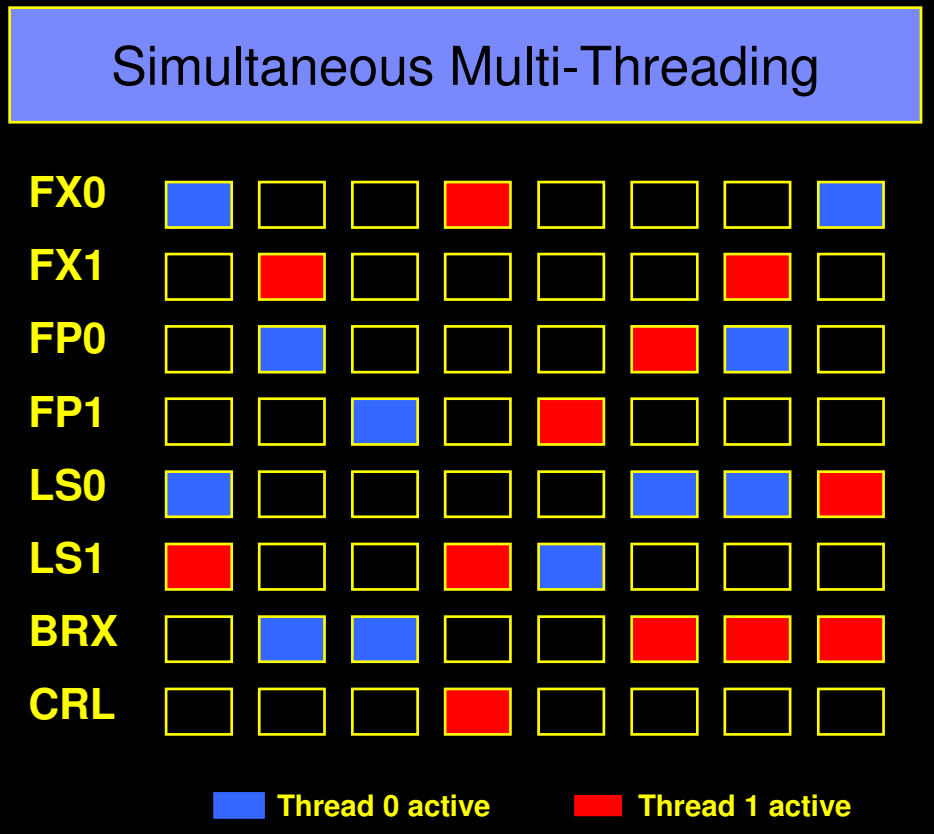
Thread 0 Executing

Thread 1 Executing

No Thread Executing

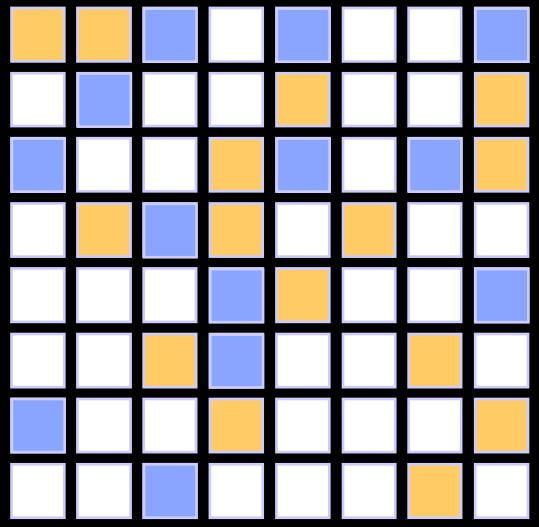
Simultaneous Multi-Threading in POWER5

- Each chip appears as a 4-way SMP to software
  - Processor resources optimized for enhanced SMT performance
- Software controlled thread priority
  - Dynamic feedback of runtime behavior to adjust priority
- Dynamic switching between single and multithreaded mode



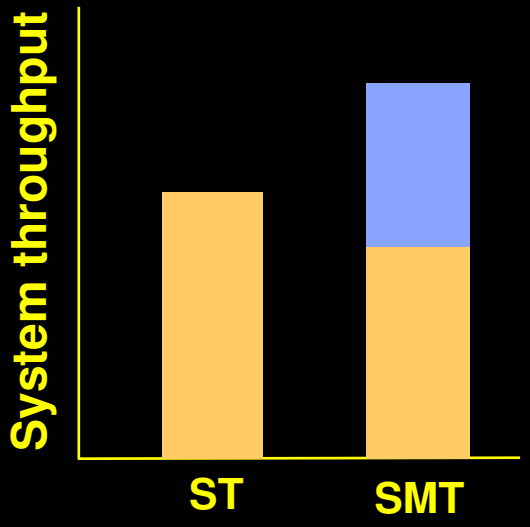
Simultaneous multi-threading

POWER5 Simultaneous Multi Threading



- Thread0 active
- No thread active
- Thread1 active

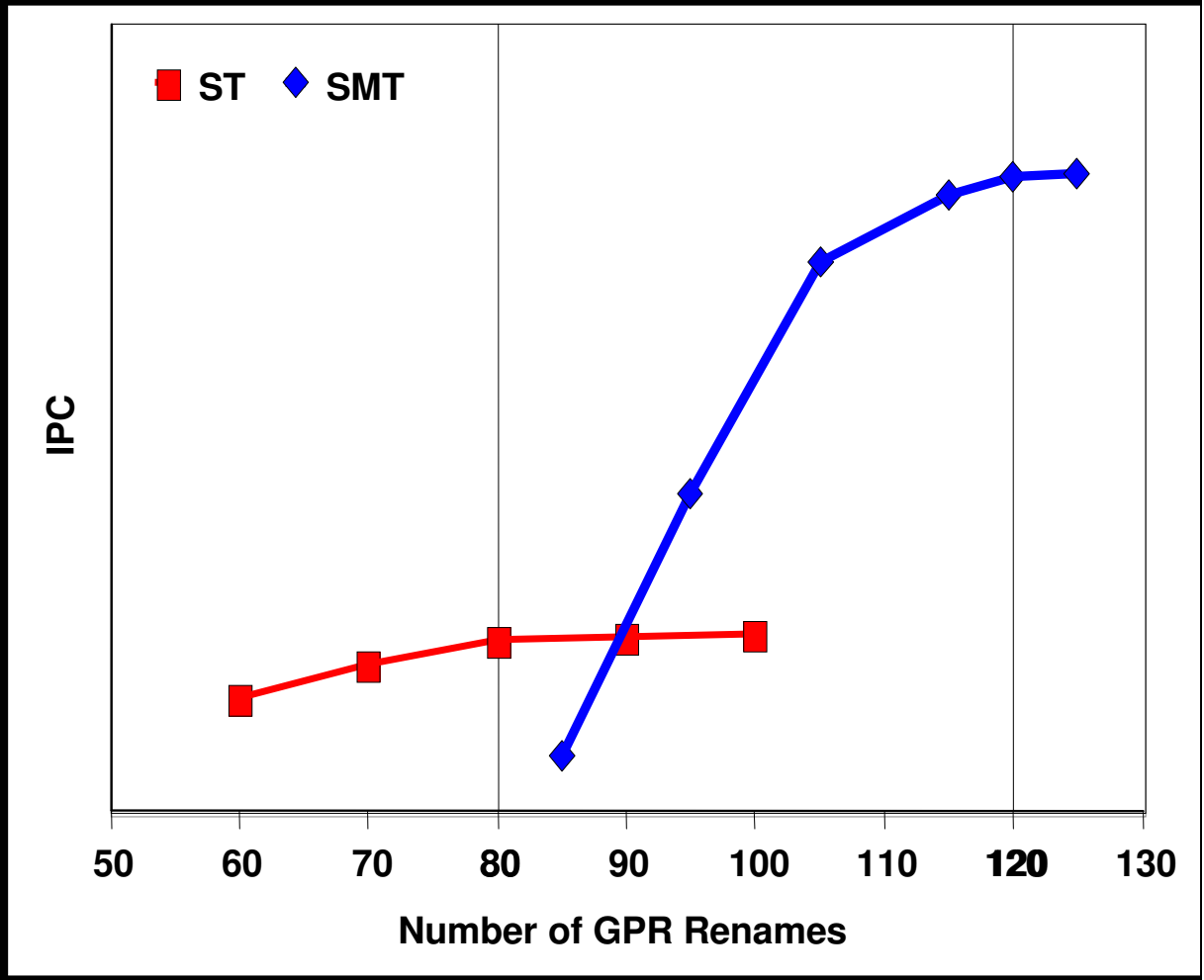
Appears as 4 CPUs per chip to the operating system (AIX 5L V5.3 and Linux)



- Utilizes unused execution unit cycles
- Symmetric multiprocessing (SMP) programming model
- Natural fit with superscalar out-of-order execution core
- Dispatch two threads per processor. Net result:
  - Better processor utilization

Resource Sizes

- Analysis done to optimize every micro-architectural resource size
  - GPR/FPR rename pool size
  - I-fetch buffers
  - Reservation Station
  - SLB/TLB/ERAT
  - I-cache/D-cache
- Many Workloads examined
- Associativity also examined



Results based on simulation of an online transaction processing application

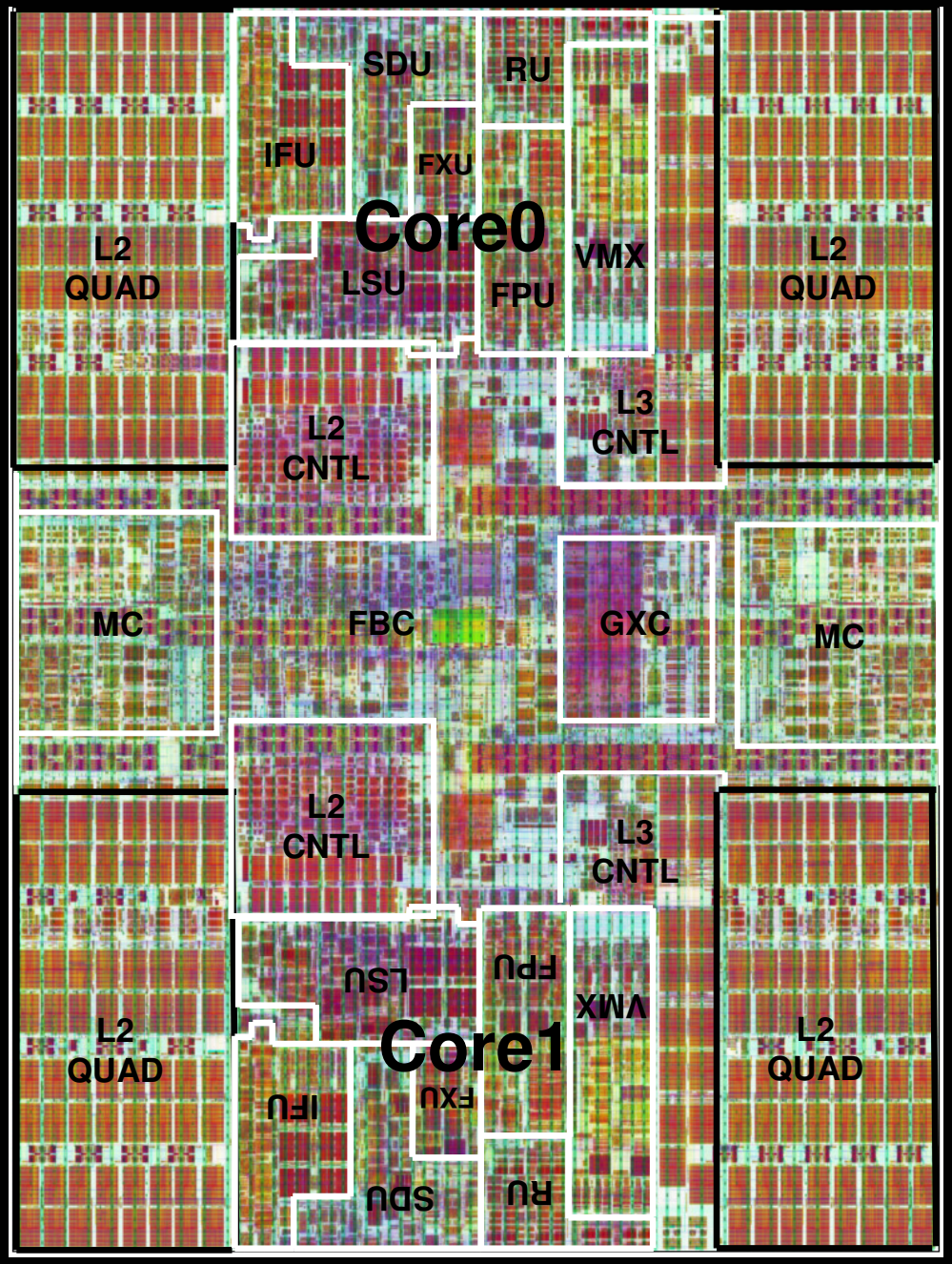
Vertical axis does not originate at 0

POWER6 Objectives

- **Processor Core**
  - High single-thread performance with ultra high frequency (13FO4) and optimized pipelines
  - Higher instruction throughput: improved SMT
- **Cache and Memory Subsystem**
  - Increase cache sizes and associativity
  - Low memory latency and increased bandwidth
- **System Architecture**
  - Fully integrated SMP fabric switch
  - Predictive subspace snooping for significant reduction of snoop traffic
  - Higher coherence bandwidth
  - Excellent scalability
- **Ultra-high frequency buses**
  - High bandwidth per pin
  - Enables lower cost packaging
- **Power**
  - Minimize latch count
  - Dynamic Power management

POWER6 Chip

- Ultra-high frequency dual-core chip
  - 7-way superscalar, 2-way SMT core
    - up to 5 instr. for one thread, up to 2 for other
  - 8 execution units
    - 2LS, 2FP, 2FX, 1BR, 1VMX
  - 790M transistors, 341 mm^2 die
  - Up to 64-core SMP systems
  - 2x4MB on-chip L2 – point of coherency
  - On-chip L3 directory and controller
  - Two memory controllers on-chip
- Technology
  - CMOS 65nm lithography, SOI Cu
- High-speed elastic bus interface at 2:1 freq
  - I/Os: 1953 signal, 5399 Power/Gnd
- Full error checking and recovery



Processor Design

	POWER5+	POWER6
Style	General out-of-order execution	Mostly in-order with special case out-of-order execution
Units	2FX, 2LS, 2FP, 1BR, 1CR	2FX, 2LS, 2FP, 1BR/CR, 1VMX
Threading	2 SMT threads Alternate ifetch Alternate dispatch (up to 5 instructions)	2 SMT threads Priority-based dispatch Simultaneous dispatch from two threads (up to 7 instructions)

POWER5+ and POWER6 Storage Hierarchy

	POWER5+	POWER6
L1 Cache		
ICache capacity, associativity	64 KB, 2-way	64 KB, 4-way
DCache capacity, associativity	32 KB, 4-way	64 KB, 8-way
L2 Cache		
Capacity, line size	1.9 MB, 128 B line	2 x 4 MB, 128 B line
Associativity, replacement	10-way, LRU	8-way, LRU
Off-chip L3 Cache		
Capacity, line size	36 MB, 256 B line	32 MB, 128 B line
Associativity, replacement	12-way, LRU	16-way, LRU
Memory	4 TB maximum	8 TB maximum
Memory bus	2x DRAM frequency	4x DRAM frequency

Network

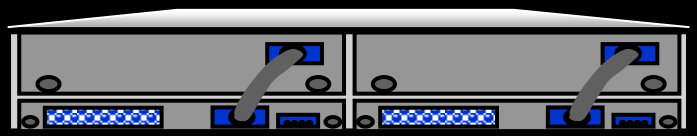
Switch Technology

- Internal network
  - In lieu of Gigabit Ethernet, Myrinet, Quadrics, etc.
  - Fourth generation
  - HPS Switch (POWER2 generation)
  - SP Switch (POWER2 -> POWER3)
  - SP Switch 2 (POWER3 -> POWER4)
  - HPS (POWER4 -> POWER5)
- Multiple links per node
  - Typically 2 links per node
  - Recently available: 1 link per node

High Performance Switch (HPS)

- Also known as “Federation”
- Follow on to SP Switch2
  - Also known as “Colony”
- Specifications:
  - 2 Gbyte/s (bidirectional)
  - < 5 microsecond latency
- Configuration:
  - Up to four adaptors per node
    - 1 or 2 links per adaptor
    - Up to 16 Gbyte/s per node

HPS Specifications



	Year	Latency [microsec.]	Bandwidth, single [Mbyte/s]	Bandwidth, multiple [Mbyte/s]
TPMX	1998	25	125	125
SP Switch 2	2001	15	350	550
HPS	2003	4.7	1800	3300

Software Overview

- Operating System
  - AIX
- Compilers
  - C
  - C++
  - Fortran
- Batch queue scheduler
  - LoadLeveler (IBM)
  - LSF (Platform)
  - PBS
  - Gridware

AIX

- Current Version: AIX 5.3
- Processors:
  - POWER3
  - POWER4
  - POWER5
- Linux Affinity
- Logical PARtitions (LPAR) Nodes
  - Operating system
  - Memory
  - Network connections
- Kernel Address Size:
  - 64-bit
  - 32-bit

Linux on POWER

- Native Linux, SLES9, RHA4
- Rpm's and package managers
- Cluster Systems Manager
- 64-bit kernel

Compiler	User Name
C	xlc,xlc_r
C++	xlC,xlC_r
Fortran	xlf, xlf90_r

Compilers

C and C++

- XL C and C++ Professional for AIX
  - Versions 6, 7, 8
  - ANSI C options
  - C++
- Compiler names:
  - xlc
  - xlC

Fortran

- XL Fortran for AIX
  - Versions 8, 9, 10
    - Fortran 77
    - Fortran 90/95/2003
- Compiler names:
  - xlf77
  - xlf90
  - xlf95

Compiler Names

Compiler	User Name
Fortran 77	xlf
Fortran 90	xlf90
C	xlc
C++	xlC
MPI compile	mpxlf, mpcc
Reentrant	xlf_r, xlc_r

AIX uses different compiler names to perform some tasks which are handled by compiler flags on many other systems

Compiler Usage

Language	Command	Feature	Extension
ANSI C	xlc xlc_r	ANSI Thread safe	.c
Extended C	cc	Pre-ANSI	.c
MPI, C	mpcc	MPI	.c
C++	xlC xlC_r	Thread safe	.C .cc .cpp
Fortran 77	xlf xlf_r	Thread safe	.f
Fortran 90	xlf90 xlf90_r	Thread safe	.f
MPI fortran	mpxlf	MPI	.f

User Limits

- Set by the system administrator
- ulimit:
  - C or K shell built-in
  - Sets or reports resource limits
  - Limits are defined in /etc/security/limits
    - Sizes are in 512 byte blocks
    - Times are in seconds
  - /usr/bin/ulimit is available from any shell

ulimit Defaults

		Value	
Limit	Definition	Default	Typical
fsize	File Size	2097151	Unlimited (-1)
core	Core File Size	2097151	Unlimited (-1)
cpu	Per Process limit	-1 (unlimited)	Unlimited (-1)
data	Data Segment Size	262144	Unlimited (-1)
stack	Stack Segment Size	65536	*Unlimited (-1)
No. files	File Descriptor Limit	2000	2000

* 64-bit address mode

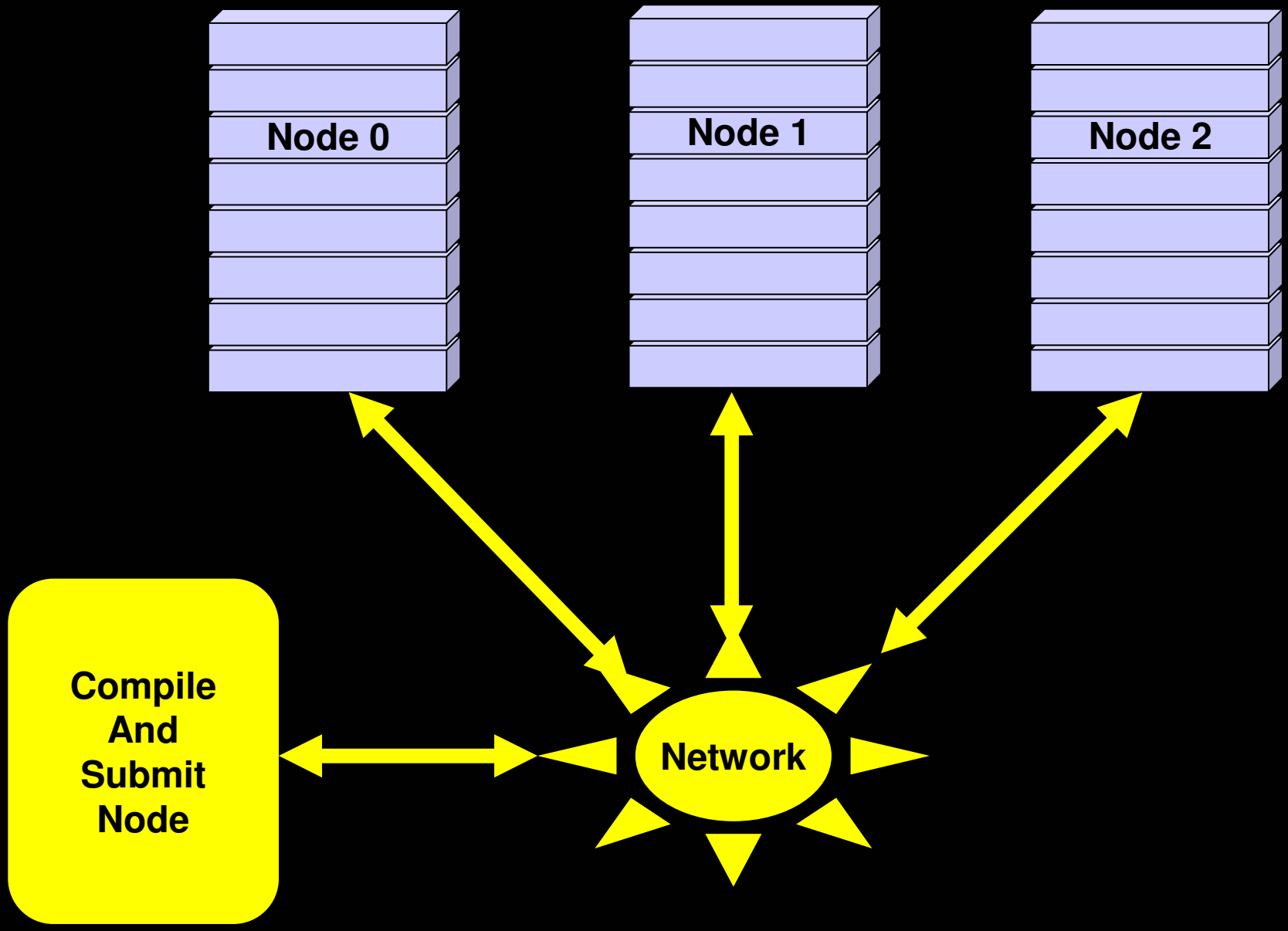
Other Environment Variables

- Thread control
  - May be set for default in /etc/environment
    - AIXTHREAD_SCOPE=S
    - AIXTHREAD_MNRATIO=1:1
    - AIXTHREAD_COND_DEBUG=OFF
    - AIXTHREAD_GUARDPAGES=4
    - AIXTHREAD_MUTEX_DEBUG=OFF
    - AIXTHREAD_RWLOCK_DEBUG=OFF

Batch Queuing

- Compile on any AIX node
  - Use –qarch=pwr5 –qtune=pwr5
- Submit job with available batch utility
- Use appropriate queue name
- Available queuing systems:
  - LoadLeveler
  - PBS
  - Gridware
  - LSF

Cluster Layout



Batch Queue: LoadLeveler

- Job Management
  - Build, Submit, Schedule, Monitor
- Workload Balancing
  - Resource scheduling
- Control
  - Centralized system administration

LoadLeveler Cluster

- Central Manager
  - Central resource manager and workload balancer, but not a central point of failure
- Execute Node
  - Runs work (serial job steps or parallel job tasks) dispatched by the Central Manager
- Scheduler Node (public or local)
  - Manages jobs from submission through completion
- Submit-only Node
  - Submits jobs to LoadLeveler from outside the cluster. Runs no daemons.

LoadLeveler Commands

- llsubmit - submits a job to LoadLeveler
- llcancel - cancels a submitted job step
- llq - queries the status of jobs in the queue
- llstatus - queries the status of machines in the cluster
- llclass - returns information about available classes
- llprio - changes the user priority of a job step

LoadLeveler: Sequential Job

```
#!/bin/ksh
#
# @ error   = Error
# @ output  = Output
# @ notification = complete
# @ notify_user = myuid@someplace.com
# @ wall_clock_limit = 01:30:00
# @ job_type = serial
# @ class = small
# @ queue

for i in a b c
do
  a.out < input.$i > Output.$i
done
```

LoadLeveler: Parallel MPI Job

```
#!/bin/ksh
#
# @ error   = Error
# @ output  = Output
# @ notification = complete
# @ notify_user = myuid@someplace.com
# @ wall_clock_limit = 01:30:00
# @ job_type = parallel
# @ node = 1
# @ tasks_per_node = 16
# @ class = small
# @ queue

poe a.out
```

LoadLeveler: Parallel OpenMP Job

```
#!/bin/ksh
#
# @ error   = Error
# @ output  = Output
# @ notification = complete
# @ notify_user = myuid@someplace.com
# @ wall_clock_limit = 01:30:00
# @ job_type = serial
# @ resources = consumableCpus(4)
# @ class = small
# @ queue

export OMP_NUM_THREADS=4
a.out
```

Submit, Monitor, Cancel

```
$ llsubmit LL
llsubmit: The job "v60n129.pbm.ihost.com.4173" has
  been submitted.
….

$ llq
Id                       Owner      Class
  Running On
------------------------ ---------- -----------
  -----------------
v60n129.4172.0    myuid  small                v60n129

…

$ llcancel v60n129.4172.0
llcancel: Cancel command has been sent to the
  central manager.
```

Documentation

- Software:
  - http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp
- Compilers
  - /usr/vac/doc/en_US/pdf
  - /usr/vacpp/doc/en_US/pdf
  - /usr/lpp/xlf/doc/en_US/pdf
- Redbooks:
  - www.redbooks.ibm.com/
    - IBM eServer p5 590 and 595 System Handbook